**Experiment-4**

**Aim**

Queries using Conversion functions(to\_char,to\_number and to\_date) , string functions(Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next\_day, add\_months, last\_day, months\_between, least, greatest, trunc, round, to\_char, to\_date).

**Description**

## Conversion Functions

Conversion functions are used to convert data from one data type to another. These are particularly useful when handling input/output operations or when data needs to be processed in a specific format.

**Common Conversion Functions:**

* **int() / to\_int()**: Converts a value to an integer.
* **float() / to\_float()**: Converts a value to a floating-point number.
* **str() / to\_string()**: Converts a value to a string.
* **bool()**: Converts a value to a boolean (True/False).
* **datetime()**: Converts a string or timestamp into a date/time object.

**String Functions**

String functions allow manipulation and analysis of string data. They are used to extract, modify, or analyze textual data.

**Common String Functions:**

* **length() / len()**: Returns the length of a string.
* **substring() / substr()**: Extracts a portion of a string.
* **concat() / +**: Concatenates two or more strings.
* **upper() / toUpperCase()**: Converts a string to uppercase.
* **lower() / toLowerCase()**: Converts a string to lowercase.
* **replace()**: Replaces occurrences of a substring with another.
* **trim()**: Removes whitespace from the beginning and end of a string.
* **split()**: Splits a string into a list based on a delimiter.

**Date Functions**

Date functions are used to manipulate and analyze date and time values. They are crucial for handling scheduling, timestamps, and time-based data.

**Common Date Functions:**

* **now() / current\_timestamp()**: Returns the current date and time.
* **date()**: Extracts the date part from a datetime value.
* **time()**: Extracts the time part from a datetime value.
* **add\_days() / date\_add()**: Adds a specific number of days to a date.
* **datediff()**: Calculates the difference between two dates.
* **format\_date()**: Formats a date into a specified pattern.
* **strftime()** (Python): Converts a date to a string based on format codes.

**Procedure**

**Conversion functions**

### (i)TO\_CHAR :It Converts a **number** or **date** to a **string**. It is Useful for formatting numbers and dates for display purposes.

**Syntax**:

TO\_CHAR(value, format)

* value: The number or date to be converted.
* Format(optional): A format string to specify how the result should look.

**Examples**:

### (i)select to\_char(6789) from dual;

### (ii)select to\_char(sysdate) from dual;

### (iii)select to\_char(sysdate,’mm-dd-yyyy’) from dual;

### (ii) TO\_NUMBER: It converts a **string** to a **number**. It is often used when data is stored in string form but needs to be treated numerically.

**Syntax**:

TO\_NUMBER(value, format)

* value: The string to be converted.
* format (Optional): A format model that matches the expected input string.

**Examples**:

**Converting a string to a number:**

SELECT TO\_NUMBER('12345.67') FROM dual;

* **With a specific format:**

SELECT TO\_NUMBER('12,345,600.67', '99,999,999.99') FROM dual;

### (iii) TO\_DATE: It is used to Converts a **string** to a **date**. It is Commonly used for converting user input (like a string) into a date format that the database can work with.

**Syntax**:

TO\_DATE(value, format)

* value: The string to be converted.
* format: The date format of the input string.

**Examples**:

**Converting a string to a date:**

SELECT TO\_DATE('10/OCT/2024', 'DD/MON/YYYY') FROM dual;

**Converting a string with time to a date:**

SELECT TO\_DATE('2024/10/10 11:30:00', 'YYYY/MM/DD HH:MI:SS') FROM dual;

**2.** **string functions**

### (i) ****Concatenation****

**Syntax**: select concat(string1,string2) from tablename;

**Example**:

(i)select concat(‘dbms’,’subject’) from dual;

(ii) select concat(roll\_number,name),marks from student;

### (ii)****Left Padding:** The LPAD is used to add characters to the left up to a given length.**

**Syntax**:**LPAD(string, length, pad\_string)**

**Example**:SELECT LPAD('SQL', 6, '\*') FROM dual;

### ****(iii)Right Padding:** The RPAD is used to add characters to the right up to a given length.**

### **Syntax**:RPAD(string, length, pad\_string)

**Example**:SELECT RPAD('SQL', 6, '\*') FROM dual;

### ****(iv)Left Trim:** The LTRIM r**emoves a specified set of characters from the beginning (left) of a string.

**Syntax**: LTRIM(string, trim\_chars)

**Example**:

1. SELECT LTRIM('\*\*\*SQL', '\*') FROM dual;
2. Select roll\_number,ltrim(name,’tha’) from student;

### ****(v)Right Trim** It re**moves a specified set of characters from the end (right) of a string.

**Syntax**:**TRIM(string, trim\_chars)**

**Example**:

1. SELECT RTRIM('SQL\*\*\*', '\*') FROM dual;
2. Select rtrim roll\_number,rtrim(name,’avs’);

**(vi)LOWER:** It converts all characters in a string to lowercase.

1. **Syntax**:**LOWER(string)**
2. **Example**:

(i)SELECT LOWER('SQL Tutorial') FROM dual;

(ii) select roll\_number,marks,lower(name) from student;

### (vii) UPPER: It converts all characters in a string to uppercase.

**Syntax**:**UPPER(string)**

**Example**:

(i)SELECT UPPER('sql tutorial') FROM dual;

(ii) select roll\_number,upper(name) from student;

### (viii)INITCAP: It converts the first letter of each word to uppercase, and the rest to lowercase.

**Syntax**:**INITCAP(string)**

**Example**:

(i)SELECT INITCAP('sql tutorial') FROM dual;

(ii) select roll\_number,initcap(name) from student;

### (ix)LENGTH: It returns the length (number of characters) of a string.

**Syntax**:**LENGTH(string)**

**Example**:

1. SELECT LENGTH('SQL Tutorial') FROM dual;
2. Select roll\_number,length(roll\_number),name,length(name) from student;

### (x) ****Substring:** It** Extracts a substring from a string starting from a specified position and with a specified length.

**Syntax**:**SUBSTR(string, start\_position, length)**

**Example**:

(i)SELECT SUBSTR('SQL Tutorial', 5, 7) FROM dual;

(ii)Select roll\_number,substr(name,2,2) from student;

### (xi)INSTR ****(In-String Search):** It** Returns the position of the first occurrence of a substring within a string.

**Syntax**: **INSTR(string, substring, start\_position, occurrence)**

**Example**:

(i) SELECT INSTR('SQL Tutorial', 't',2,1) FROM dual;

**(xii)Translate:** The TRANSLATE function in SQL is used to replace each character in a string with its corresponding character from another string.

**Syntax : select translate(string,from char,to char) from tablename**

**Example:**

**(i)Select translate(‘abaaabbaabnhf’,’a’,’b’) from dual;**

**(ii)Select roll\_number,translate(name,’t’,’q’) from student;**

**Date functions**

### (i) ****SYSDATE:** It will display** the current date and time of the system.

* **Syntax**: **SYSDATE**
* **Example**: SELECT SYSDATE FROM dual;

### (ii) **NEXT\_DAY: It will display the** date of the next specified day of the week after a given date.

* **Syntax**: **NEXT\_DAY(date, weekday)**
* **Example**:

**(i)**SELECT NEXT\_DAY(SYSDATE, 'MONDAY') FROM dual;

**(ii) select next\_day(’06-jan-2025’,’mon’) from dual;**

### **(iii)**ADD\_MONTHS:** It will** Add a specified number of months to a given date.

* **Syntax**: **ADD\_MONTHS(date, n)**

**Example**:

(i)SELECT ADD\_MONTHS(SYSDATE, 2) FROM dual;

(ii) select roll\_number,join\_date,add\_months(join\_date,2) from student;

### (iv) **LAST\_DAY: It will display** the last day of the month for the specified date.

* **Syntax**: **LAST\_DAY(date)**
* **Example**:

**(i)**SELECT LAST\_DAY(SYSDATE) FROM dual;

**(ii)** select roll\_number,join\_date,last\_day(join\_date) from student;

### (v)****MONTHS\_BETWEEN:** It will r**eturn the number of months between two dates.

**Syntax**: **MONTHS\_BETWEEN (date1, date2)**

**Example**:

(i)SELECT MONTHS\_BETWEEN(SYSDATE,to\_date('2024-01-01','yyyy-mm-dd')) FROM dual;

**(ii)select roll\_number,join\_date,months\_between(sysdate,join\_date) from student;**

### (vi)****LEAST:** It will** Returns the least (earliest) of the provided dates.

**Syntax: LEAST(date1, date2, ...)**

**Example**:

(i)SELECT LEAST(SYSDATE, to\_date('2024-01-01','yyyy-mm-dd')) FROM dual;

1. Select least (20,45,6,78) from dual;

(vii) **GREATEST: It will** Returns the greatest (latest) of the provided dates.

* **Syntax**: **GREATEST(date1, date2, ...)**
* **Example**:

**(i)**SELECTGREATEST(SYSDATE, '2024-01-01') FROM dual;

**(ii) select greatest (65,7,89,90) from dual;**

### (viii) ****TRUNC: It will**** Truncates the date to the specified unit (year, month, day, etc.).

**Syntax**: **TRUNC(date, [format])**

**Example**:

### (i) sql> select trunc(to\_date('10-5-1990','dd-mm-yyyy'),'dd') from dual;

### output : trunc(to\_

### ---------

### 10-may-90

### (ii) sql> select trunc(to\_date('10-5-1994','dd-mm-yyyy'),'yyyy') from dual;

### output: trunc(to\_

### ---------

### 01-jan-94

### (ix) ****ROUND: It will**** Rounds the date to the nearest unit specified.

**Syntax**: ROUND(date, [format])

**Example**:

**SQL> SELECT round(to\_date('30-5-1990','dd-mm-yyyy'),'mm') from dual;**

**OUTPUT : ROUND(TO\_**

**---------**

**01-JUN-90**

### ****(x)TO\_CHAR: It**** Converts a date to a string with a specified format.

**Syntax**: TO\_CHAR(date, format)

**Example**:SELECT TO\_CHAR(SYSDATE, 'YYYY-MM-DD HH:MI:SS') FROM dual;

### (xi)****TO\_DATE: It**** Converts a string to a date using a specified format.

**Syntax**: TO\_DATE(string, format)

**Example**:SELECT TO\_DATE('2024-10-13', 'YYYY-MM-DD') FROM dual;